ISSN 0027-1349, Moscow University Physics Bulletin, 2024, Vol. 79, Suppl. 2, pp. S647–S665. © Allerton Press, Inc., 2024.

Enhanced Image Clustering with Random-Bond Ising Models Using LDPC Graph Representations and Nishimori Temperature

V. S. Usatyuk[1,2*], D. A. Sapozhnikov[1], and S. I. Egorov[2]

[1] *Research and Development Department, T8 LLC, Moscow, 107076 Russia*

[2] *Department of Computer Engineering, South-West State University, Kursk, 305040 Russia*

Received October 1, 2024; revised October 10, 2024; accepted October 20, 2024

Abstract—This paper addresses the challenge of improving clustering accuracy of image data, particularly focusing on feature representations extracted from convolutional deep neural networks (CNNs). Traditional spectral clustering methods often struggle with high dimension features tensors generated by CNNs like the VGG model. To overcome these limitations, this work proposes a novel approach that enhances spectral clustering by utilizing sparse graph representations (hyperbolic embedding) based on quasi-cyclic low-density parity check (QC-LDPC) and multiedge type (MET) QC-LDPC codes. These graphs are constructed using progressive edge growth (PEG), simulated annealing methods. The paper tackles the specific problem of effectively clustering high-dimensional, sparse image features by modeling their interactions with a random-bond Ising model (RBIM). The optimization process leverages Nishimori temperature estimation to assign weights to graph edges based on image features, leading to more accurate grouping of images into distinct clusters. This approach can be applied to various tasks, including classification. The proposed method not only improves clustering accuracy but also reduces the number of required parameters. It achieves a 17.39% improvement in accuracy (90.60%) compared to state-of-the-art Erdős–Rényi graphs (73.21%), which lack the hardware-efficient structure of QC-LDPC graphs. By utilizing sparse feature parameters, an efficient MET QC-LDPC multigraph is created that outperforms conventional techniques such as mean-field approximation and Laplacian methods in graph clustering, binary classification. These findings highlight the potential of this approach for a wide range of applications, including image clustering, neural network pruning, data representation, and neuron activation pattern prediction.

Keywords: clustering, contextual sparsity, DNN, mean-field, multiedge type graph, nishimori, pruning, random bond Ising model, quasi-cyclic low-density parity-check codes, QC-LDPC, sparse graph

DOI: 10.3103/S0027134924702102

1. INTRODUCTION

Graph clustering is critical in machine learning tasks such as image classification, pattern recognition, and data mining. Numerous methods, many rooted in statistical physics, have been developed to tackle this problem, including Ising spin glass spectral clustering [1, 2], mean field approximation (MFA) [3], the Laplacian Method [4, 5], and Nishimori temperature spectral clustering [6, 7]. These techniques focus on partitioning graph nodes into clusters that reveal meaningful data patterns.

Recent advancements in hyperbolic embedding have shown promise, particularly for large-scale graphs with hierarchical or treelike structures, commonly seen in image data. Hyperbolic spaces, with their exponential expansion, allow for efficient modeling of these complex graphs, enhancing tasks like image classification by leveraging hierarchical relationships [8, 9]. However, the application of hyperbolic models to sparse, high-dimensional data, such as features produced by convolutional neural networks (CNNs) like VGG [10], remains a challenge.

On the other hand, supervised methods like ResNet-50 and EfficientNet, which utilize more advanced CNNs compared to VGG [11, 12], as well as visual transformers [13, 14] and hyperbolic visual hierarchy encoding [8], still struggle to achieve high classification accuracy. These approaches often re-

*E-mail: L@Lcrypto.com

quire computationally expensive training, particularly in the case of visual transformers.

Treelike low-density parity-check (LDPC) codes offer a compelling alternative for clustering due to their scale-free structure, small-world properties, and higher clustering coefficient. Unlike Erdös—Rényi (ER) graphs, which exhibit Poisson degree distributions [7], LDPC graphs can model power-law degree distributions, reflecting the presence of "superpixel hub" nodes—an important feature in image data. The higher clustering coefficient in LDPC graphs also enables more efficient clustering through dense local node interconnections, making them suitable for handling sparse feature spaces.

LDPC codes, typically known for their use in error-correcting codes, provide sparse graph structures with high connectivity but low density, making them ideal candidates for hyperbolic embedding and spectral clustering. By combining LDPC graphs with hyperbolic embeddings, we can create a multigraph structure that preserves sparsity while capturing hierarchical relationships, leading to improved clustering performance on high-dimensional image features extracted by CNNs.

This paper explores the integration of Nishimori temperature estimation within this framework to enhance spectral clustering on LDPC-based sparse graphs. Nishimori temperature, derived from the random bond Ising model (RBIM), optimizes clustering by tuning the system at a critical phase transition between ordered (ferromagnetic) and disordered (paramagnetic) states [15, 16]. When applied to LDPC graph representations, this approach ensures robust and efficient clustering of sparse, high-dimensional image data.

In summary, we propose a novel method that integrates hyperbolic embedding, LDPC graph construction, and Nishimori temperature estimation within the RBIM spectral clustering framework. This approach significantly improves clustering accuracy and reduces computational complexity by leveraging the strengths of hyperbolic geometry, LDPC sparsity, and RBIM optimization.

The following sections delve into the theoretical foundations of the random bond Ising model, the Bethe–Hessian matrix, and Nishimori temperature, as well as key concepts from hyperbolic embedding and LDPC graph theory.

2. RANDOM BOND ISING MODELS

The Ising model is a fundamental tool in statistical physics used to study the magnetic properties of materials. It posits that magnetic moments, or spins, interact solely with their nearest neighbors. This model can be generalized to arbitrary graphs, providing a versatile framework for analyzing complex systems.

Definition 1. RBIM. Consider a $\zeta(\nu, \varepsilon)$-graph, where ν represents the set of vertices corresponding to spins, and ε represents the set of edges corresponding to interactions between spins. Each edge (i, j) is assigned a weight J_{ij}, which determines the interaction strength between spins i and j. The spin vector $s = \{-1, 1\}^n$ on the graph is a random vector with a Boltzmann distribution:

$$\mu(s) = \frac{e^{-\beta \mathcal{H}_{\mathcal{J}}(\mathbf{s})}}{Z_{J,\beta}},$$

where β is a positive number, $Z_{J,\beta}$ is the normalization coefficient, and the Hamiltonian of the Ising model on the graph ζ is given by:

$$\mathcal{H}_{\mathcal{J}}(\mathbf{s}) = -\sum_{(ij)\in\varepsilon} J_{ij} s_i s_j = -s^T J s.$$

In the Ising model, the goal is to find a configuration of spins s that minimizes the Hamiltonian $\mathcal{H}_{\mathcal{J}}(\mathbf{s})$. The clustering problem on graphs can be transformed into an optimization problem of the Ising model by assigning weights to the graph edges based on vertex similarities. For instance, in the case of binary classification of graph vertices, the weight of the edge (i, j) can be defined as:

$$J_{ij} = \begin{cases} 1, & \text{if} \quad s_i = s_j \\ -1, & \text{if} \quad s_i \neq s_j. \end{cases} \tag{1}$$

This corresponds to an Ising model of the ζ-graph where spins corresponding to vertices in the same cluster interact with a positive force, and spins corresponding to vertices in different clusters interact with a negative force.

The RBIM energy-based model (EBM) uses a Gibbs–Boltzmann density [17]:

$$p(x;\theta) = \frac{1}{Z(\theta)} \exp\{-U(x;\theta)\},$$

where $x \in \mathbb{R}^D$ represents an image signal, $U(x;\theta)$ is a (convolutional, ConvNet [18]) neural network with weights θ and a scalar output, and $Z(\theta)$ is the intractable normalizing constant. Training involves approximating $p(x;\theta^*) \approx q(x)$ using i.i.d. samples from the data distribution $q(x)$. Such models allow for the encoding of images using ground states of Ising models.

2.1. Nishimori Temperature and Phase Transition

The Nishimori temperature is the critical temperature at which a phase transition occurs in the Ising model. This temperature is crucial in the study of spin glasses and has significant applications in optimization and machine learning. For the Ising model on a graph, the Nishimori temperature is defined as $T_N = \frac{1}{\beta_N}$. Consider the following definition, [7]:

Definition 2. Let $\zeta(\nu, \varepsilon)$ be a graph, where ν is the set of vertices and ε is the set of edges, and let J be a matrix corresponding to the graph. The nonzero elements of the matrix are generated according to the rule:

$$P(x) = p(|x|)e^{\beta_N x},$$

where $p(|x|)$ is a nonnegative function satisfying the normalization condition $\int p_0(|x|)e^{\beta_N x}\,dx = 1$, and $\beta_N > 0$ is the inverse of Nishimori temperature.

For example, in the Edwards–Anderson model for normally distributed weights in graph vertex classification problems, β_N is defined as :

$$P(x) = \frac{1}{\sqrt{2\pi\nu^2}} \exp\left\{-\frac{(x-J_0)^2}{2\nu^2}\right\},$$
$$p_0(|x|) = \frac{1}{\sqrt{2\pi\nu^2}} \exp\left\{-\frac{x^2-J_0^2}{2\nu^2}\right\},$$
$$\beta_N = \frac{J_0}{\nu^2}, J_0 = \mathbb{E}[J], \mathrm{Var}[J] = \nu. \qquad (2)$$

Above the Nishimori temperature, the Ising model enters a paramagnetic phase where all spins are uncorrelated and randomly oriented. Below the Nishimori temperature, the model transitions into a spin glass state, where the spins are frozen in configurations corresponding to local minima of the Hamiltonian. Under the Nishimori temperature, let $\mathrm{A} \in \{0,1\}^{\mathrm{n}\times\mathrm{n}}$ be the symmetric adjacency matrix of $\zeta(\nu,\varepsilon)$, defined by $\mathrm{A_{ij}} = 1$ if $(\mathrm{ij}) \in \varepsilon$, and $\mathrm{A_{ij}} = 0$ otherwise. Let $\mathrm{D} \in \mathbb{N}^{\mathrm{n}\times\mathrm{n}}$ be the diagonal degree matrix $\mathrm{D} = \mathrm{diag}(\mathrm{A1_n})$, easily obtains that, [7]:

$$\mathbb{E}\left[\mathrm{H}_{\beta_\mathrm{N},\mathrm{J}}\right] = \mathrm{I_n} + \mathbb{E}\left[\frac{\tanh(\beta \mathrm{J_{ij}})}{1-\tanh^2(\beta \mathrm{J_{ij}})}\right](\mathrm{D}-\mathrm{A}).$$

By varying the temperature T, we can observe a phase transition, as illustrated in Fig. 1. In the context of graph clustering, the Nishimori temperature represents the phase transition point between a regime where vertices are uncorrelated and a regime where vertices cluster based on their similarities.

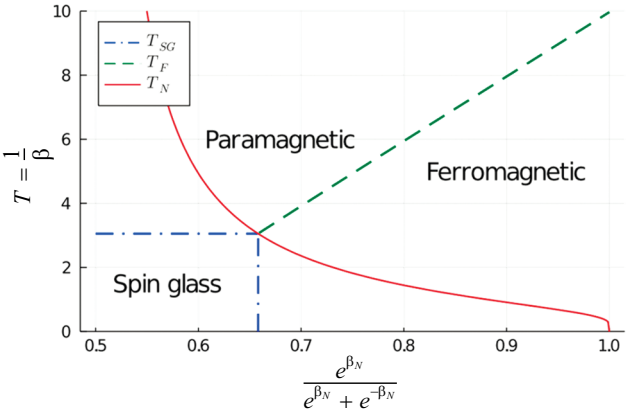

Fig. 1. Phase diagram of the RBIM for $J_{ij} \in \{-1, 1\}$. The x-axis ranges from $\frac{1}{2}$ for $\beta_\mathrm{N} = 0$ to 1 for $\beta_\mathrm{N} \to \infty$. The y-axis represents T, the inverse of β. The dashed green line corresponds to the inverse of β_F, the dash-dotted blue line to the inverse of β_SG, and the solid red line to the inverse of β_N, [7].

2.2. Bethe Energy and Bethe–Hessian

Let us introduce the free energy $F_{J\beta}$ for $\mu(s)$ in the Boltzmann distribution and its Bethe approximation $\widetilde{F}_{J\beta}(q)$ for a set of parameters q:

$$F_{J\beta} = \sum_s \mu(s)\left(\beta H_J(s) + \ln \mu(s)\right),$$
$$\widetilde{F}_{J\beta}(q) = \sum_s p_q(s)\left(\beta H_J(s) + \ln p_q(s)\right),$$

where s denotes a particular state of the system, $\mu(s)$ is the marginal probability of the system being in state s according to the Boltzmann distribution.

The free energy $F_{J\beta}$ generates moments of the Boltzmann distribution (Ising), but it cannot generally be calculated exactly. The variational Bethe free energy $\widetilde{F}_{J\beta}$ is an approximation. Minimizing $\widetilde{F}_{J\beta}$ with respect to q for a parameterized family of distributions $p_q(s)$ yields a minimum of the Kullback–Leibler divergence for the optimal approximation of the free energy. The mean and covariance-parameterized distribution $p_q = p_{m,\chi}$ is defined as follows:

$$p_{m,\chi} = \prod_{(ij)\in\varepsilon} \frac{1 + m_i s_i + m_j s_j + \chi_{ij} s_i s_j}{4},$$
$$\prod_{i=1}^{n} \left[\frac{1+m_i s_i}{2}\right]^{1-d_i},$$

where m_i and χ_{ij} are the averages of s_i and s_{ij} by distribution, and $d_i = |\{j : (ij) \in \varepsilon\}|$ is the degree of the ith node.

Combining the expressions for free energy and its approximation, we find that $\widetilde{F}^{\mathrm{Bethe}}_{J\beta}(m,\chi)\big|_{m=0} = 0$,

indicating that the paramagnetic point always delivers a minimum of Bethe energy.

$$\left.\frac{\partial^2 \widetilde{F}^{\text{Bethe}}_{J\beta}}{\partial m_i \partial m_j}\right|_{m=0} = \delta_{ij}\left(1+\sum_{k\in\partial i}\frac{\chi_{ik}^2}{1-\chi_{ik}^2}\right)-\frac{\chi_{ik}}{1-\chi_{ik}^2},$$

where δ_{ij} is the Kronecker delta, which is 1 if $i=j$ and 0 otherwise, $k\in\partial i$ refers to the set of nodes (or variables) k that are neighbors of the node i in a graphical model or network. This set is often called the "neighborhood" of i. The expression $\sum_{k\in\partial i}$ means that the summation is taken over all neighbors k of node i. The first term involves a sum over the neighbors k of node i, accounting for the contributions from each neighbor to the diagonal part of the Hessian matrix $H_{\beta,J}$. The second term represents the off-diagonal elements, which are associated with the interactions between different nodes i and j. In summary, $k\in\partial i$ denotes all the nodes that are directly connected to node i in the graph, and these neighboring nodes contribute to the calculation of the Hessian in the context of the Bethe approximation.

By calculating the gradient of the Bethe energy with respect to χ and assuming $\chi_{ij}=\tanh(\beta J_{ij})$, we obtain an expression for the Hessian approximation:

$$H_{\beta,J}=\delta_{ij}\left(1+\sum_{k\in\partial i}\frac{\tanh^2(\beta J_{ik})}{1-\tanh^2(\beta J_{ik})}\right)-\frac{\tanh(\beta J_{ij})}{1-\tanh^2(\beta J_{ij})}.$$

The Hessian matrix $H_{\beta,J}$ approximates the second derivative (or curvature) of the energy function with respect to the parameters. The given expression is an approximation that results from applying the Bethe approximation. The Kronecker delta ensures that the summation and the additional constant 1 only affect the diagonal elements, while the off-diagonal elements are influenced only by the second term.

2.3. Relationship between Nishimori Temperature and Bethe–Hessian

We can now establish a connection between the Nishimori temperature and the Bethe–Hessian matrix. It has been shown that at the Nishimori point β_N, the smallest eigenvalue of the matrix $H_{\beta,J}$ is zero. This property can be used to estimate the Nishimori temperature by finding the smallest eigenvalue of the matrix $H_{\beta,J}$ as a function of β and choosing the value of β for which the smallest eigenvalue is zero (within some specified accuracy):

$$\widehat{\beta}_N=\max_{\beta}\left(\beta:\gamma_{\min}(H_{\beta,J})=0\right),$$

where $\gamma_{\min}$ corresponds to the smallest eigenvalue.

This property also allows us to associate the phase transition in the graph Ising model with the smallest eigenvalue of the matrix $H_{\beta,J}$. As mentioned earlier, in the high-temperature phase, the system is paramagnetic and the matrix $H_{\beta,J}$ is positive definite. As the temperature decreases, the system can transition to the ferromagnetic or spin-glass phase. At the phase transition point, the smallest eigenvalue of the matrix $H_{\beta,J}$ becomes zero. This means that the phase transition point corresponds to the (inverse) Nishimori temperature β_N. With all necessary definitions established, we can now move directly to clustering the data on the graph.

2.4. RBIM under Sparse Graphs

Sparse graphs can be utilized as $\zeta(\nu,\varepsilon)$-graphs for the random bond Ising model (RBIM). In this section, we explore Erdös–Rényi graphs, LDPC progressive edge growth (PEG) graphs, and quasi-cyclic simulated annealing graphs.

Erdös–Rényi (ER) graphs, as introduced in [19], provide a flexible framework for modeling random networks. An ER graph $\zeta(\nu,\varepsilon)$=$\mathcal{G}(n,p)$ consists of n vertices where each pair of vertices is connected by an edge with probability p. On average, such a graph contains $\binom{n}{2}p$ edges. The degree distribution of vertices in an ER graph follows a binomial distribution:

$$P(\deg(v)=k)=\binom{n-1}{k}p^k(1-p)^{n-1-k},$$

where n represents the total number of vertices. ER graphs offer several advantages for our application: they enable controllable sparsity through the parameter p, allowing us to tailor model complexity based on desired feature reduction levels. Moreover, their well-understood theoretical properties and inherent randomness contribute to both interpretability and robust generalization performance.

Definition 3. LDPC (low-density parity-check) codes are linear block error-correcting codes denoted as $[N,K]$, characterized by sparse parity-check matrices $H^{(N-K)\times N}$. The matrix H specifies parity check equations and can be visualized as a Tanner graph. For example, the Tanner graph (Fig. 2, left) corresponding to the parity-check matrix:

$$H=\begin{bmatrix}1&0&1&1&1\\1&1&0&0&0\\0&1&1&1&1\end{bmatrix}.$$

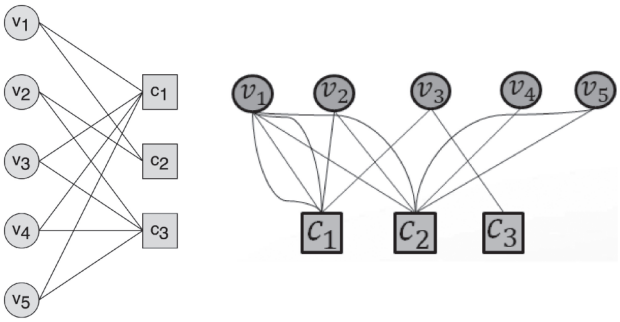

Fig. 2. (left) Tanner graph of H parity-check matrix and (right) multigraph protograph of H_2 parity-check matrix.

Definition 4. Quasi-cyclic low-density parity-check (QC-LDPC) codes form a subclass with a quasi-cyclic parity-check matrix H [20]. In an (N, K) QC-LDPC code, N is the code length (number of codeword bits), K is the number of message bits, and the remaining bits are parity bits. The Tanner graph of a QC-LDPC code is described by the parity-check matrix $H^{mL\times nL}$, consisting of square blocks of size $L \times L$ that are either zero matrices or circulant permutation matrices (CPM).

The $L \times L$ circulant permutation matrix P is defined as:

$$P_{ij} = \begin{cases} 1, & \text{if} \quad i+1 \equiv j \mod L \\ 0, & \text{otherwise.} \end{cases}$$

The P_k represents the CPM (circularly) shifting the identity matrix I to the right by k times for any k, $0 \le k \le L-1$. Denote the set $\{0, 1, \ldots, L-1\}$ by A_L. Suppose the matrix H_{QC} of size $mL \times nL$ is defined as:

$$H_{QC} = \begin{bmatrix} P_{a_{11}} & P_{a_{12}} & \cdots & P_{a_{1n}} \\ P_{a_{21}} & P_{a_{22}} & \cdots & P_{a_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ P_{a_{m1}} & P_{a_{m2}} & \cdots & P_{a_{mn}} \end{bmatrix},$$

where $a_{ij} \in A_L$, and L is the circulant size of H_{QC}. The exponent matrix $E(H)$ is derived from H_{QC} with assigned place shift values $a_{i,j}$, and the protograph matrix $M(H_{QC})$ is obtained by placing 1 for nonzero circulant entries and 0 otherwise.

A QC-LDPC code can be represented as a multigraph, as shown in Fig. 2 (right), using the parity-check matrix H_2:

$$H_2 = \begin{pmatrix} I_1 + I_2 + I_7 & I_9 & I_{23} & 0 & 0 \\ I_{12} + I_{37} & I_{19} & 0 & I_{32} & I_{11} + I_{12} \\ 0 & 0 & I_{33} & 0 & 0 \end{pmatrix},$$

where I sum denotes the CPM with weight > 1 used in multi-edge (MET) QC-LDPC codes. MET QC-LDPC codes are typically defined by variable and check node degree distributions [21].

If there is a cycle of length $2l$ in the Tanner graph of $M(H)$, it is called a *block-cycle* of length $2l$. Any block-cycle in $M(H)$ of length $2l$ corresponds both to the sequence of $2l$ CPM's $P_{a_1}, P_{a_2}, \ldots, P_{a_{2l}}$ in H and sequence of $2l$ integers $a_1, a_2, ..., a_{2l}$ in $E(H)$ which will be called *exponent chain*. The Tanner graph exhibits cycles of size l based on the equation

$$\sum_{i=1}^{2l} (-1)^i a_i \equiv 0 \mod L. \tag{3}$$

Trapping sets within the Tanner graph, formed by cycles (block-cycle for QC-LDPC) or cycle (block-cycle) overlap, consist of a variable nodes and b odd-degree checks, denoted as a trapping set $TS(a, b)$. The minimum codeword weight (Hamming distance), determining the code distance ($d_{\min}$), corresponds to $TS(a, 0)$, where $a = d_{\min}$. Trapping sets give rise to pseudocodewords [22, 23].

The metric extrinsic message degree (EMD) of a cycle in the Tanner graph is defined as the number of check nodes singly connected to the variable nodes involved in the cycle. The EMD value of a code is an important characteristic, because each cycle is a trapping set. The EMD metric estimates how strongly subgraph of cycle is connected with the rest of the Tanner graph. Calculation of the Tanner graph EMD is a hard task, because it requires to determine if the edge is extrinsic edge or cut edge. Instead of EMD, approximated cycle EMD (ACE) can be used, [24, 25]. It is easy to calculate by equation:

$$ACE(C) = \sum_{v \in E(V_c)} (d(v) - 2),$$

where C-cycles in Tanner graph, $d(v)$ is the degree of the variable node participated in cycle, V_c—the set of variable nodes in cycle. In [26] proofs that for a LDPC code with girth g, any cycles of length $2l$ has equal ACE and EMD if $l < g - 2$. For example for code with girth 6 for all cycles 6, ACE metric estimation gives the EMD value. It means that fast calculation ACE gives the same result as EMD. Increasing the minimum values (for fixed cycle length) of EMD (ACE) values effectively eliminates TS. This means that the detrimental impact of a trapping set $TS(a, b)$ depends solely on its harm value, calculated as $harm = \frac{b}{a}$. Consequently, under this restriction, focusing on minimizing both the probability of error within subgraphs and the multiplicity of these harmful trapping sets is sufficient to construct graphs resilient

against the most damaging types of suboptimal solutions. In (binary) synapse neural networks, optimization requires a careful balance between treelike pseudocodewords ($TS(a,b)$) and Hamming distance codewords ($d_{\min} = a$, $TS(a,0)$), aiming to maximize storage capacity α. This trade-off is illustrated in Figs. 3 and 4, as discussed in [27].

Research demonstrates that slow, dynamic aging in glassy systems can spontaneously generate structures resembling equilibrium spin glass solutions [2]. These aging glasses exhibit complex history-dependent behaviors—deviating from standard thermodynamic equilibrium. This phenomenon is often described as a shift from "power-limited" to "bandwidth-limited regimes" in information theory, highlighting the system's increased capacity to store and process information [28]. A defining characteristic of these systems is their "memory"—the ability to retain past configurations. This suggests they explore a larger configuration space with more than two states (beyond simple spin-up or spin-down). The q-ary Potts model, an extension of the Ising model allowing q different spin values, provides a valuable framework for understanding this expansion. Applying the principles of the q-ary Potts model opens exciting possibilities. We can explore sophisticated graph models like nonbinary LDPC graphs (RBIM), which better capture the intricate dynamics of memory and aging seen in glassy systems and related nonbinary synapse neural networks. One key advantage of these q-ary RBIM is their ability to break cycles more effectively than binary counterparts, leading to improved capacity through increased code distance $TS(a,0)$ and enhanced capacity (error correction), [29]. This cycle breaking capability stems from the careful selection of graphs q-ary variables.

3. RBIM LDPC GRAPH CONSTRUCTION

Increasing $d_{\min} = a$ enhances the network's capacity as a ratio of the number of distinguishable feature vectors to the number of nonzero parameters (number of neurons) to a larger minimum Hamming distance. However, this can also complicate the process of finding the optimal solution due to a smaller and more constrained target region. On the other hand, expanding the treelike structure enlarges the region of the target extremum but decreases the distance between local minima. Thus, achieving the right balance is crucial for optimizing the network's performance and robustness. In the original progressive edge-growth (PEG) algorithm [30], a code is constructed based on a variable node degree sequence. This sequence is determined in advance using the number of variable nodes N and the variable node degree sequence $D_v \in \mathbb{Z}^N$. The variable node degree sequence specifies the number of nonzero elements in each column of the parity-check matrix.

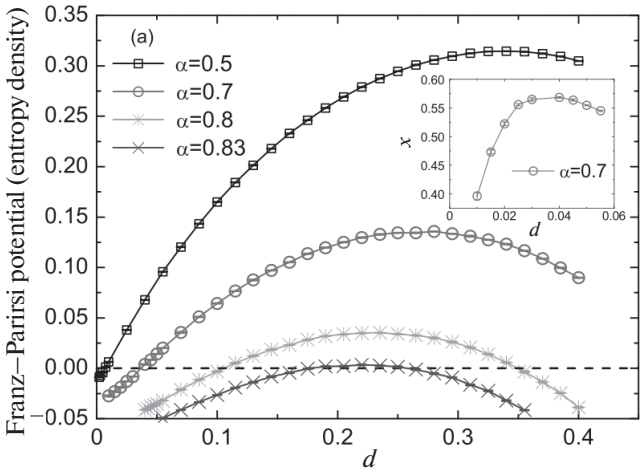

Fig. 3. Franz–Parisi potential plotted against the normalized Hamming distance. The inset shows the coupling field's behavior with respect to distance for $\alpha = 0.7$. The observed maximum indicates a change in the concavity of the entropy curve, a feature also present for other finite values of α [27].

A PEG-based algorithm involves two fundamental procedures: local graph expansion and check node selection. These procedures are executed sequentially to construct a Tanner graph that connects symbol and check nodes edge by edge. During the local graph expansion, a symbol node's neighborhood is expanded to detect and avoid short cycles when adding new edges. Check nodes that would create cycles are pruned, and if avoiding a cycle is impossible, only a subset of candidate check nodes producing the largest possible cycle remains. The check node selection procedure then reduces this list of candidate nodes based on the current graph structure. In typical PEG algorithms, this procedure aims to balance the check node degrees by selecting candidates with the lowest check node degree using breadth first search (BFS). The extrinsic message degree (EMD) and approximate cycle EMD (ACE) [24, 25] are commonly used metrics to assess the connectivity of variable nodes.

The algorithm proposed in this paper enhances the PEG algorithm by maximizing ACE (PEG + ACE maximization), as detailed in Algorithm 1, further we will apply the abbreviation PEG, implementation available [31]. For the construction of QC-LDPC graphs and MET QC-LDPC multigraphs, we propose using simulated annealing method with EMD and Hamming distance maximization, [32].

The detailed steps of this iterative process are presented in Algorithm 2. The construction of QC-

Algorithm 1. Progressive edge growth (PEG) with ACE maximization

Input: Number of variable nodes N, number of check nodes M, variable node v degree sequence $D_v \in \mathbb{Z}^N$.
Output: Parity-check matrix $H \in \{0,1\}^{M\times N}$.

```
Initialize the matrix H ← 0_{M×N}.
for v ← 1 to N do
  for k ← 1 to D_v(v) do
    if k = 1 then
      Find c ∈ C, C = {c | Σ_j H_{c,j} is minimum}.
      Randomly select check node c* ∈ C.
      Set H_{c*,v} ← 1.
      Update auxiliary arrays
    else
      Initialize BFS tree with root node v.
      Perform BFS expansion from node v.
      while Tree expansion is possible do
        Mark visited variable and check nodes
        Add unvisited v to the current level
        if ∄v which added ¬∀c are included then
          Find check node c* with min. weight
          Randomly select among candidates
          Set H_{c*,v} ← 1
          break
        end if
        Add unvisited c connected to new v
        if All check nodes are included then
          Select check node c* that max ACE.
        else
          Randomly select c* from candidates
        end if
        Set H_{c*,v} ← 1
        break
        Move to the next tree level
      end while
    end if
  end for
end for
  return H
```

Algorithm 2. Simulated annealing method for construction of QC and MET LDPC (multi)graphs

Input: $M(H)$—protograph matrix, L—circulant size, g—girth of constructed matrix, EMD—minimal EMD value, $Iter$—maximal number of iterations, $seed$—a seed to be used in a pseudo-random number generator, $Temp$—initial value of temperature.
Output: Quasi-cyclic parity-check matrix $E(H)$.

$Nstep = 0$
$i, j = rnd(seed)$
for $it = 0; it <= Iter; it = it + 1$ **do**
  while $M_{ij}(H) = 0$ **do**
    $i, j = rnd(seed)$
  end while
  for $k = 0; k <= L-1; k = k+1$ **do**

$$\Theta_k = enumcirccycles(i, j, k, g, EMD),$$
$$w(k) = e^{\frac{-\Theta_k}{Temp}},$$
$$P(k) = w(k) / \sum_{m=0}^{L-1} w(m),$$

where Θ_k is the number of cycles through $E_{ij}(H)$-CPM with shift value k, $P(k)$ is the probability of k-shift CPM value choice, $w(k)$ is the probability weight function.

  end for

$$\Phi = enumcycles(E(H), g, EMD),$$

where Φ is the total number of cycles in $E(H)$.

$$E_{ij}(H) = rndshift(P, Temp).$$

$Nstep = Nstep + 1.$

$$Temp = \eta \frac{\Phi}{Nstep^2},$$

where η is some constant value.
end for
  return $E(H)$

LDPC codes begins by generating an initial exponent matrix. This is done by assigning suboptimal random shift values to the nonempty CPMs, as outlined in [20]. Since no optimization is applied at this stage, the resulting matrix often exhibits undesirable structural properties, such as low girth, the presence of short cycles in the code's Tanner graph, and a low EMD value. These suboptimal characteristics adversely affect the capacity of the resulting QC-LDPC graphs. To mitigate these issues, a two-stage approach is employed:

First stage: Improve of initial exponent matrix. The process starts with the reconstruction of an exponent matrix using Eq. (3) and a greedy algorithm designed to minimize (or eliminate) the number of short cycles by carefully selecting CPM shift values. However, due to the limitations of the greedy method and the suboptimal initial matrix, the resulting matrices often still have low girth and low EMD value.

Second stage: Simulated annealing optimization. To further refine the exponent matrix obtained in the first stage, a simulated annealing optimization technique is applied. This metaheuristic algorithm begins with a relatively high "temperature" parameter, allowing for exploration of a wide range of solutions, including those that may temporarily worsen matrix properties (such as increasing short cycles). As the temperature gradually decreases, the focus shifts to refining the solution by reducing short cycles and optimizing the overall cycle structure.

The primary objective of this process is to generate an exponent matrix with improved girth and EMD, leading to a QC-LDPC graph with enhanced DNN capacity capabilities. Finally, a method for estimating

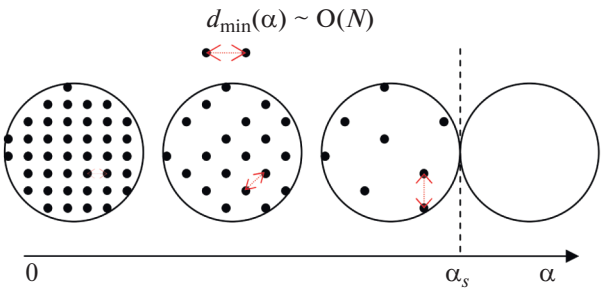

Fig. 4. Schematic representation of the weight space. Points denote equilibrium solutions for weights. $\alpha \approx 0.833$ represents the storage capacity for binary synapses (after which, the solution space becomes empty), $d_{\min}$ denotes the actual Hamming distance without normalization, [27].

and increasing the Hamming distance is applied to further optimize the QC and MET QC-LDPC, [33].

The proposed PEG algorithm with ACE maximization and the simulated annealing (SA) method with EMD maximization have already been applied to factorization problems under Ising spin glass energy-based models (EBM) on dense graphs, network issues, surface meshes, and covariance matrices. These approaches demonstrate significant improvements in reconstruction accuracy using the Frobenius norm—up to 8 orders of magnitude in individual cases—as shown in [33]. The source code for the proposed algorithms and their application for clustering, along with publicly available data, is published on GitHub [34].

4. CLUSTERING USING RBIM

To evaluate the efficiency of our clustering method, we examine two distinct graph problems: classifying graph vertices in the Edwards–Anderson model and image clustering. We compare the clustering outcomes for three graph types: Erdös–Rényi random graphs, PEG graphs, and quasi-cyclic graphs, keeping all other parameters constant. Specifically, for both problems, the average vertex degree and the number of vertices are held the same. Figure 5 shows the trace of the adjacency matrices for ER, PEG, QC graphs. These graphs are constructed with an average degree of $(c = 15)$ and include $(n = 6000)$ vertices, reflecting a typical setup for an image clustering problem

We benchmark the proposed clustering method against methods utilizing spin glass temperature, mean field, and Laplacian matrix clustering. Below is a formal description of these methods, which rely on assigning each data element to an eigenvector component corresponding to the minimum eigenvalue:

1. Bethe–Hessian matrices at Nishimori temperature (Bethe–Hessian Nishimori clustering):

$$(H_{\beta_N,J}) = \delta_{ij}\left(1 + \sum_{k\in\partial i}\frac{\tanh^2(\beta_N J_{ik})}{1-\tanh^2(\beta_N J_{ik})}\right) - \frac{\tanh(\beta_N J_{ij})}{1-\tanh^2(\beta_N J_{ij})}. \quad (4)$$

2. Bethe–Hessian matrices at spin glass temperature (Bethe–Hessian spin clustering):

$$(H_{\beta_{SG},J}) = \delta_{ij}\left(1 + \sum_{k\in\partial i}\frac{\tanh^2(\beta_{SG} J_{ik})}{1-\tanh^2(\beta_{SG} J_{ik})}\right) - \frac{\tanh(\beta_{SG} J_{ij})}{1-\tanh^2(\beta_{SG} J_{ij})}.$$

3. Laplacian matrices:

$$(L_J) = \delta_{ij}\left(\sum_{k\in\partial i}|J_{ik}|\right) - J_{ij}.$$

4. Weighted matrix of the graph J.

Eigenvectors derived from the adjacency matrix analysis are clustered using the k-nearest neighbors (KNN) method [35, 36]. This yields two distinct clusters. Subsequently, KNN is employed to classify novel data points based on their proximity to these established clusters. Figure 6 illustrates the distribution of eigenvalues for ER graphs, PEG graphs, and QC (multi)graphs. These eigenvalues correspond to the spectrum of matrix B in the complex plane. Because the weighted adjacency matrix is asymmetric, its eigenvalues are complex numbers.

In comparison to Erdös-Rényi random graphs, both PEG and QC graphs exhibit more concentrated eigenvalue distributions with pronounced ground state structures. This characteristic suggests superior clustering performance for these graph types.

Definition 4. Weighted nonbacktracking matrix. Given a graph $\zeta(\nu,\varepsilon)$ and a function $f:\varepsilon\to\mathbb{R}$ assigning weights ω_e to edges e, the weighted nonbacktracking matrix $B\in\mathbb{R}^{2|\varepsilon|\times 2|\varepsilon|}$ is defined on the set of directed edges of $\mathcal{G}$ as:

$$B_{(ij),(k\ell)} = \delta_{jk}(1-\delta_{i\ell})\omega_{k\ell}. \quad (5)$$

Definition 5. Bethe–Hessian matrix given a graph $\zeta(\nu,\varepsilon)$, a function $f:\varepsilon\to\mathbb{R}$ with $f(e)=\omega_e$ for all $e\in\varepsilon$, and a parameter $x\in\mathbb{C}\setminus\{\pm\omega_{ij}\}_{(ij)\in\varepsilon}$, the Bethe–Hessian matrix $H(x)\in\mathbb{C}^{n\times n}$ is defined as:

$$H_{ij}(x) = \left(1+\sum_{k\in\partial i}\frac{\omega_{ik}^2}{x^2-\omega_{ik}^2}\right)\delta_{ij} - \frac{x\omega_{ij}}{x^2-\omega_{ij}^2}.$$

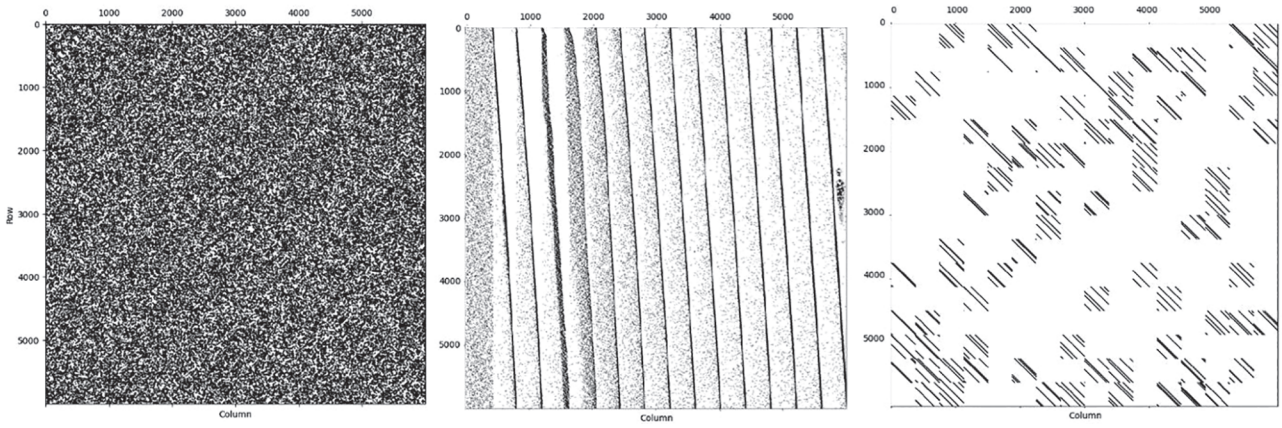

Fig. 5. Trace of graph adjacency matrices from left to right: Erdös–Rényi, PEG, QC.

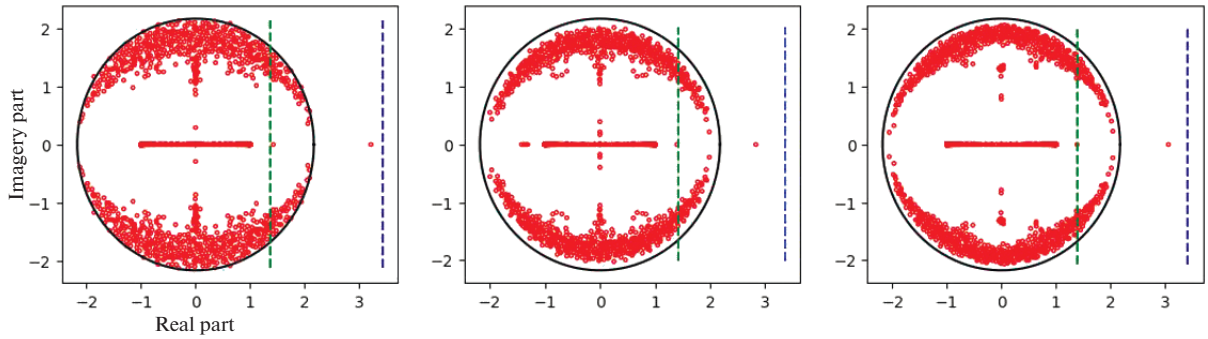

Fig. 6. Spectrum of the matrix B in the complex plane. The entries J_{ij} are generated independently according to $\mathcal{N}(J_0, \nu^2)$. The weights appearing in Eq. (5) are defined as $\omega_{ij} = \tanh(\beta J_{ij})$. Sparse regime: $n = 500$, $c = 5$, $J_0 = 1$, $\nu = 1$, $\beta = 10$. For all plots, the dashed blue line corresponds to $c\mathbb{E}[\tanh(\beta J)]$, the dash-dotted green line to $\mathbb{E}[\tanh^2(\beta J)]/\mathbb{E}[\tanh(\beta J)]$, while the solid black line represents the circle in the complex plane centered at the origin with radius $\sqrt{c\mathbb{E}[\tanh^2(\beta J)]}$. From left to right: Erdös–Rényi, PEG, QC. The abscissa x-axis represents the real part, and the ordinate y-axis represents the imaginary part.

Since $\zeta(\nu, \varepsilon)$ is undirected, $H(x)$ is symmetric but not Hermitian unless $x \in \mathbb{R}$.

Watanabe–Fukumizu equation, [37–39]. Let $H(x)$ and B be matrices on the same graph $\zeta(\nu, \varepsilon)$ with the same weighting function f. For $x \in \mathbb{C} \setminus \{\pm\omega_{ij}\}_{(ij)\in\varepsilon}$:

$$\det[xI_{2|\varepsilon|} - B] = \det[H(x)] \prod_{(ij)\in\varepsilon} (x^2 - \omega_{ij}^2).$$

If x is in the spectrum of B, $\det[H(x)] = 0$.

To achieve accurate node clustering within our RBIM framework on sparse graphs, we leverage the Nishimori temperature estimation algorithm proposed in [7], Alg. 3. Precise knowledge of the Nishimori temperature, denoted as β_{N}, is crucial for obtaining a reliable estimate of the true node classes, represented by σ. A robust estimator for σ can be derived from the signs of the entries within the eigenvector $\mathbf{x}$ corresponding to the smallest eigenvalue (close to zero) of the Bethe–Hessian matrix $\mathrm{H}_{\beta_{\mathrm{N}},\tilde{\mathrm{J}}}$.

To perform accurate node clustering, knowledge of the Nishimori temperature β_{N} is essential for obtaining a precise estimate of the true node classes $\hat{\sigma}$, Alg. 4. A powerful estimator of $\hat{\sigma}$ is derived from the signs of the entries of the eigenvector x of the Bethe–Hessian matrix $H_{\beta_{\mathrm{N}},\tilde{J}}$, associated with its smallest amplitude eigenvalue, which is close to zero. This estimator is particularly effective in sparse graphs, where traditional clustering methods may fail to accurately capture the underlying community structure.

In the following, we use the *overlap* as a measure to compare the inference performance of various node classification algorithms based on spectral clustering. The overlap is defined as:

$$\text{Overlap} = \left| 2\left(\frac{1}{n} \sum_{i=1}^{n} \delta_{\sigma_i, \hat{\sigma}_i} - \frac{1}{2} \right) \right|, \qquad (6)$$

where $\hat{\sigma}_i$ represents the estimated label of node i. The overlap value ranges from 0 (indicating a random assignment) to 1 (indicating a perfect assignment).

Algorithm 3. Nishimori temperature estimation

Input: Weighted adjacency matrix of a graph $\mathbf{J} \in \mathbb{R}^{n\times n}$, precision error $\varepsilon \in \mathbb{R}$.

Output: Estimated Nishimori temperature $\hat{\beta}_{\rm N} \in \mathbb{R}^{+}$.

1. Compute c, the average degree of the underlying unweighted graph: $c = \frac{1}{n}\sum_i \sum_j \mathbb{I}(\mathbf{J}_{ij} \neq 0)$.
2. Compute $\hat{\beta}_{\rm SG}$ by solving $c\mathbb{E}[\tanh^2(\hat{\beta}_{\rm SG}\mathbf{J}_{ij})] = 1$.
3. Set $t = 1$ and $\beta_t \leftarrow \hat{\beta}_{\rm SG}$.
4. Initialize $\delta \leftarrow +\infty$.

   while $\delta > \varepsilon$:

   (a) Compute $\mathbf{H}_{\beta_t,\mathbf{J}}$, Eq. (4).

   (b) Compute $\gamma_{\min,t}$, the smallest eigenvalue of $\mathbf{H}_{\beta_t,\mathbf{J}}$, and its associated eigenvector $\mathbf{x}_t$.

   (c) Define the function $f_t(\beta') = \mathbf{x}_t^T\mathbf{H}_{\beta',\mathbf{J}}\mathbf{x}_t$, for $\beta' \in \mathbb{R}^{+}$.

   (d) Compute β_{t+1} by solving $f_t(\beta_{t+1}) = 0$.

   (e) Update $\delta \leftarrow |\gamma_{\min,t}|$.

   (f) Increment $t \leftarrow t + 1$.

return: β_{t-1}.

Algorithm 4. Nishimori–Bethe relation for node classification

Input: Weighted adjacency matrix of a graph $\widetilde{\mathbf{J}} \in \mathbb{R}^{n\times n}$, precision error $\varepsilon \in \mathbb{R}$.

Output: Estimated Nishimori temperature $\hat{\beta}_{\rm N} \in \mathbb{R}^{+}$, estimated label vector $\hat{\sigma} \in \{-1, 1\}^n$.

1. Shift the nonzero entries of $\widetilde{\mathbf{J}}$:
$$\widetilde{\mathbf{J}}_{ij} \leftarrow \widetilde{\mathbf{J}}_{ij} - \frac{1}{2|E|}\mathbf{1}_n^T\widetilde{\mathbf{J}}\mathbf{1}_n.$$
2. Compute $\hat{\beta}_{\rm N}$ using Alg. 3.
3. Compute $\mathbf{H}_{\beta_t,\mathbf{J}}$, Eq. (4).
4. Compute $\mathbf{x} \leftarrow$ the eigenvector associated with $\gamma_{\min}(\mathbf{H}_{\hat{\beta}_{\rm N},\widetilde{\mathbf{J}}})$.
5. Estimate $\hat{\sigma}$ as the output of 2-class k-means on the entries of $\mathbf{x}$.

return: $\hat{\beta}_{\rm N}$, $\hat{\sigma}$.

Consider the construction of an RBIM $\zeta(\nu,\varepsilon)$-graph (MET QC-LDPC multigraph) with 6000 nodes and a column weight of 10 using simulated annealing, where $\zeta(\nu,\varepsilon) = \{E(H)_1, E(H)_2\}$, Eqs. (A.1a), (A.1b). The RBIM based on $E(H)_1$ shows a clustering accuracy (overlap) of 76.46%, while the RBIM based on $E(H)_2$ achieves a clustering accuracy of 90.60%.

The constructed multigraph $E(H)_1$, Eq. (A.1a), has dimensions 4×4 with a circulant permutation matrix (CPM) of weight 5 and size 1500. It has a girth of 6 with the following EMD (Extrinsic Message Degree) values:

- Cycle 6: EMD = 6 with 120 000 cycles.
- Cycle 8: EMD = 0 with 3000 cycles, EMD = 2 with 28 500 cycles, EMD = 4 with 352 500 cycles, EMD = 6 with 1 716 000 cycles, EMD = 8 with 3 978 000 cycles.
- Cycle 10: EMD = 2 with 264 000 cycles, EMD = 4 with 4 462 500 cycles, EMD = 6 with 29 655 000 cycles, EMD = 8 with 116 850 000 cycles, EMD = 10 with 202 647 000 cycles.
- Cycles 12: EMD 0, 2 and from 4 to 12.

The upper bound on the Hamming distance from the protograph for the first three QC rows, the last three QC rows, and the first, second, and fourth QC rows, for a graph of size 3×1500 nodes, is equal to 500.

The best RBIM graph, $E(H)_2$, Eq. (A.1b), has dimensions 16×16 with a CPM of weight 2 and size 375. It has a girth of 6 with the following EMD spectrum structures:

- Cycle 6: EMD = 9 with 7875 cycles, EMD = 11 with 26 625 cycles, EMD = 13 with 18 375 cycles, EMD = 14 with 13 500 cycles, EMD = 16 with 36 375 cycles, EMD = 18 with 24 750 cycles.
- Cycle 8: EMD = 10 with 1125 cycles, EMD = 12 with 348 000 cycles, EMD = 14 with 1 282 500 cycles, EMD = 15 with 1875 cycles, EMD = 16 with 1 675 500 cycles, EMD = 17 with 432 750 cycles, EMD = 18 with 708 000 cycles, EMD = 19 with 1 684 125 cycles, EMD = 21 with 2 125 875 cycles, EMD = 23 with 972 375 cycles.
- Cycle 10: EMD = 12 with 375 cycles, EMD = 13 with 62 250 cycles, EMD = 14 with 10 875 cycles, EMD = 15 with 11 272 875 cycles, EMD = 16 with 84 000 cycles, EMD = 17 with 56 359 875 cycles, EMD = 18 with 301 875 cycles, EMD = 19 with 107 866 500 cycles, EMD = 20 with 14 818 125 cycles, EMD = 21 with 91 248 375 cycles, EMD = 22 with

73 055 250 cycles, EMD = 23 with 29 644 500 cycles, EMD = 24 with 137 562 375 cycles, EMD = 25 with 174 000 cycles, EMD = 26 with 117 641 625 cycles, EMD = 27 with 25 500 cycles, EMD = 28 with 37 609 875 cycles.

- Cycles 12: EMD from 9 to 33.

The upper bound on the Hamming distance for the protograph, estimated using method proposed in [40, 41], is 2 625 536 when excluding the first four quasi-cyclic (QC) rows. This results in a graph of similar size to the previous example (12×375). When omitting the last four QC rows instead, this upper bound reduces to 929 792.These findings clearly demonstrate that the eliminate trapping set structures (by increase EMD values) and increased Hamming distance of the multigraph RBIM $E(H)_2$ significantly enhance the capacity of the proposed graph representation. This results in an improved accuracy of 90.60% for a weight of 10, marking a significant enhancement of 19.93% over the 70.67% accuracy achieved with $E(H)_1$ quasi-cyclic hardware-friendly graphs, which can be implemented using a shift register.

Furthermore, multigraph MET QC-LDPC RBIM $E(H)_2$ exhibits a 17.39% improvement in accuracy (90.60%) compared to the Erdös–Rényi (ER) graph (73.21%), which lacks a structural, hardware-friendly QC-LDPC representation. The different column omissions correspond to varying levels of contextual sparsity in neural network weights under different input conditions. This is because different image classes exhibit distinct nonlinear temporal correlations across the image data. Increasing the Hamming distance effectively increases the capacity to represent these diverse class characteristics, [27, 42, 43].

4.1. RBIM Clustering on Synthetic Data

For the synthetic data RBIM clustering, we will use the following initial parameters: the number of graph vertices $n = 500$ and the average vertex weight $c = 5$. According to Eq. (2), when generating the graph weights J, we will vary the standard deviation ν to alter the separability of the vertex classes. The expected weight value J_0 is positive for edges connecting vertices within the same class and negative for edges connecting vertices from different classes, as defined by Eq. (1).

Figure 7 illustrates the clustering results on synthetic data based on the ratio $\frac{\beta_N}{\beta_{SG}}$, which is a function of the standard deviation in the original data. The overlap, or the proportion of correctly clustered nodes,

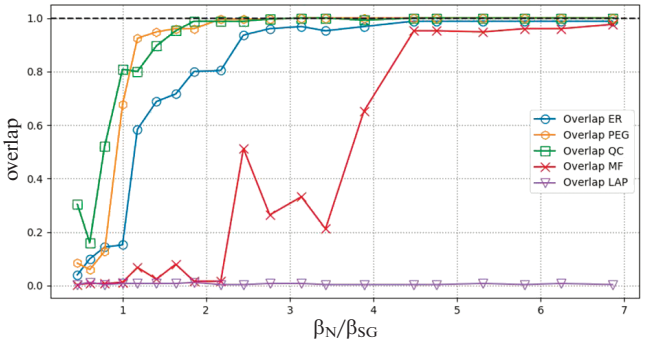

Fig. 7. Clustering accuracy (overlap, Eq. (6)) of the RBIM model on ER, PEG, and QC graphs using Mean Field and Laplacian methods. Synthetic data with $n = 500$ nodes, average column weight $c = 5$, and varying Nishimori-to-spin glass temperature ratios (β_N/β_{SG}).

is plotted on the abscissa axis. It demonstrates that the proposed modifications of the spectral clustering method using PEG and QC graphs outperform the ER graph in regions with high dispersion. It is noteworthy that as the dispersion decreases (Nishimori temperature increases), MF clustering also shows high performance. To assess the robustness of the methods against decreasing graph density, we will conduct an experiment with image data.

4.2. RBIM Clustering on GAN Images

In the image clustering task, we considered 6000 images of dogs and cats. These images were generated using a GAN model, and feature extraction was performed using the VGG16 model, [10]. Each image was represented as a vector $z_i \in \mathbb{R}^{1\times l}$, where $l = 512$ features. To introduce sparsity, each element of z_i was multiplied by a binomial random variable $y \sim \mathrm{Bin}(1, 1 - \frac{\kappa}{l})$, with $\kappa = 20$.

Utilizing sparse data offers several advantages, including reduced memory requirements, decreased computational complexity, and improved scalability. Additionally, sparse computations can help reduce energy consumption, which is crucial for developing energy-efficient computing systems. The weight of an edge connecting nodes i and j in the graph is determined using the scalar product, proposed in paper [7]:

$$J_{ij} = \frac{|(z_i, z_j)|}{l}. \qquad (7)$$

The results of image clustering for Erdös–Rényi, PEG, and QC graphs, based on the average vertex weight, are shown in Fig. 8.

The figure demonstrates that under high sparsity conditions, spectral clustering on PEG and QC

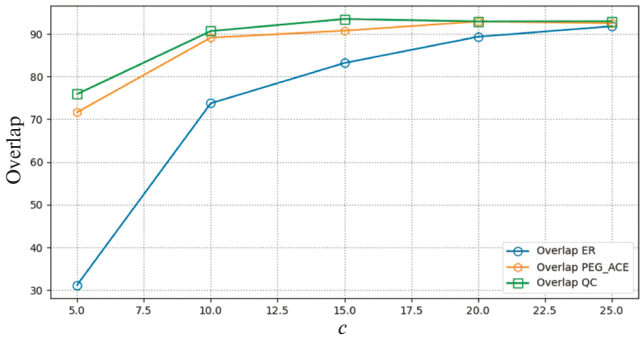

Fig. 8. Clustering accuracy (overlap, Eq. (6)) for the RBIM graph on ER, PEG, and MET QC-LDPC graphs for two image classes (cats and dogs) under varying column weights c.

graphs significantly outperforms clustering on the Erdös–Rényi graph. Even with a vertex weight of $c = 15$, an overlap of 93.4% is achieved, whereas the Erdös-Rényi graph does not reach this value even with a weight of $c = 25$.

Let's examine the case of $c = 15$ in more detail. Figures 9 present column charts for the elements of the eigenvectors corresponding to the minimum Bethe–Hessian eigenvalue for the Erdös–Rényi, PEG, and QC graphs. These elements essentially serve as labels for KNN clustering, with the proportion of elements erroneously assigned to a particular cluster shown in green.

From these figures, it is evident that with the same average vertex weight, spectral clustering on the Erdös–Rényi graph performs significantly worse in separating objects. Additionally, the clustering results for the QC graph exhibit a much smaller spread along the abscissa axis.

Next, we examine clustering across various configurations of the QC graph. The value before the comma in Table 1 shows the clustering results obtained using Eq. (7). This includes results for the spectral method at the Nishimori temperature, the spectral method at the spin glass temperature, the mean field approximation, and the Laplacian method across different scenarios.

From the data presented in the table, it is evident that the spectral method at the Nishimori temperature generally performs best for most graph configurations, achieving high percentages of correctly assigned elements. However, in cases where the calculation of the Nishimori temperature is unstable, the method's performance declines significantly. For these configurations, the Laplace method consistently shows good results, providing a reliable alternative. Specifically, the Laplace method demonstrates stability and effectiveness, particularly when the Nishimori temperature method's results are suboptimal.

We proposed more efficient cosine similarity approach compare to proposed equation (7) in paper [7]. This approach is consistent with the cosine similarity measure, which ensures faster convergence of results. On very dense graphs, convergence might be hindered due to an overly heterogeneous adjacency matrix:

$$J_{ij} = \frac{|(z_i, z_j)|}{|z_i||z_j|}. \tag{8}$$

The value following the comma in Table 1 presents the clustering results obtained using Eq. (8). This includes results for the spectral method at the Nishimori temperature, the spectral method at the spin glass temperature, the mean field approximation, and the Laplacian method across various cases.

Comparing the results in Table 1, several key differences between the original metric (Eq. (7)) and the proposed cosine similarity metric (Eq. (8)) are evident. In general, the proposed cosine similarity metric leads to improved clustering performance in many cases. This improvement is reflected in higher accuracy percentages achieved across various algorithms, including Nishimori temperature RBIM, spin glass RBIM, mean field approximation, and the Laplace method, as well as across different datasets. The cosine similarity approach ensures faster convergence of clustering results due to its alignment with the cosine similarity measure, which is particularly beneficial for large-scale graphs where computational efficiency is crucial. Additionally, this metric enhances the stability of the clustering results by reducing the sensitivity to noise and variations in the data. However, on dense graphs, the cosine similarity metric may experience reduced accuracy due to its sensitivity to the magnitude of vector components. In such cases, the large magnitudes of vectors in highly interconnected graphs can lead to smaller cosine similarities even between closely related nodes, which can decrease clustering accuracy for some graphs.

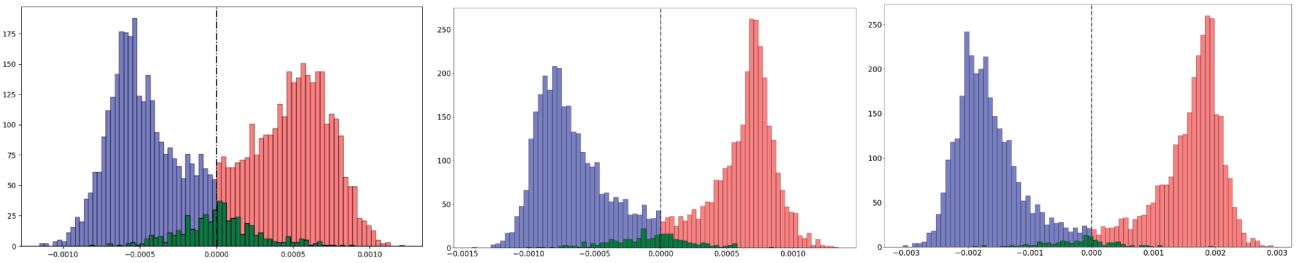

Fig. 9. Histogram of the entries of the eigenvector for spectral clustering on the graphs from left to right: Erdös–Rényi, PEG, QC.

5. CONCLUSIONS

This paper demonstrates the effectiveness of using QC and MET QC LDPC codes to construct sparse graphs, which define the bonds (graph edges, spin interactions) in RBIMs. We introduced a novel approach that leverages Nishimori temperature estimation within the QC structural RBIMs to optimize spectral clustering. Our results show that this method significantly improves clustering accuracy by exploiting the structured sparsity inherent in LDPC codes. By constructing graphs based on image features extracted from pretrained models like VGG, we were able to reduce the feature space from 512 dimensions to a sparse subset (approximately 2%) without significant loss of information. This demonstrates the ability of our approach to efficiently capture the most relevant features for distinguishing between image classes, such as binary classification of dogs and cats. The integration of Nishimori temperature estimation was crucial in optimizing the RBIM on these sparse graph representations. This technique allowed us to precisely tune the RBIM at the phase transition point, leading to more robust and accurate clustering results compared to traditional methods. Additionally, the structured sparsity offered by LDPC graphs aligns with the tendency of deep neural networks to develop sparse activations, where only a subset of neurons is engaged for a given input. This finding paves the way for more computationally efficient and interpretable models, focusing on the most informative features.

Our work also suggests that dimensionality reduction can be achieved without significant loss of information, offering a pathway to more efficient neural network architectures. The use of LDPC codes on the graph not only refines feature selection processes but also opens avenues for developing hybrid models that combine the strengths of graph-based techniques with neural network architectures. This could potentially lead to advancements in both graph theory and deep learning, particularly in the context of large-scale image classification tasks. For hardware implementations, LDPC (multi)graphs benefit from the quasi-cyclic matrix structure, which is based on shift registers, optimizing memory access and enabling parallel processing. This makes our approach not only theoretically appealing but also highly practical for real-world applications where computational efficiency is crucial.

Looking ahead, we aim to extend this research by exploring the behavior of multiclass activations in neural networks, utilizing LDPC codes to gain deeper insights into the representation of complex categorical information. By studying the relationships between nonbinary features, we hope to develop novel training strategies and further optimize feature selection. This interdisciplinary approach, combining concepts from statistical physics, graph theory, and deep learning, holds great promise for advancing our understanding of neural networks and expanding their potential across a broader range of applications.

APPENDIX

MET QC-LDPC RBIM

The following appendix includes two examples of multigraphs (MET) QC-LDPC parity-check matrices, $E(H)_1$ and $E(H)_2$ (Table 1), each consisting of 6000 nodes and a column weight of 10. The multigraph $E(H)_1$ has dimensions of 4 by 4, with a circulant permutation matrix (CPM) of size 1500 and a weight of 5. In contrast, $E(H)_2$ has dimensions of 16 by 16, with a CPM of size 375 and a weight of 2. For brevity and due to space constraints, the remaining multigraphs ($E(H)_3$ through $E(H)_{42}$ and other graphs) are not included here but can be found in the supplementary materials available at [34].

$$E(H)_1 = [E_1|E_2|E_3|E_4], \qquad \text{(A.1a)}$$

$$E(H)_2 = [E_5|E_6|E_7|E_8], \qquad \text{(A.1b)}$$

where $E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8$:

Table 1. Comparison of clustering accuracy (overlap, Eq. (6)) using different methods (Nishimori temperature RBIM, spin glass RBIM, mean-field approximation, Laplacian spectral clustering) on various QC LDPC graphs and MET QC-LDPC multigraphs. Accuracy is evaluated using both the original edge weight metric ((7), [7]) and a proposed cosine similarity metric (8)

Parity-check matrix of RBIM graphs	Nishimori	Spin glass	Mean-field	Laplacian
$E(H)_3$, size of 2×2, $L = 3000$	23.67%, 31.40%	26.63%, 30.30%	0.03%, 20.70%	32.73%, 0.03%
$E(H)_4$, size of 2×2, $L = 3000$	54.43%, 37.93%	47.00%, 32.80%	1.20%, 3.16%	0.03%, 0.10%
$E(H)_5$, size of 2×2, $L = 3000$	53.40%, 33.10%	46.57%, 33.50%	1.10%, 4.56%	0.10%, 0.03%
$E(H)_6$, size of 2×2, $L = 3000$	26.20%, 31.73%	29.10%, 26.56%	0.03%, 22.46%	31.97%, 0.03%
$E(H)_7$, size of 2×2, $L = 3000$	23.67%, 31.40%	26.63%, 30.33%	0.03%, 20.70%	32.73%, 0.03%
$E(H)_8$, size of 2×2, $L = 3000$	54.43%, 37.93%	47.00%, 32.80%	1.27%, 3.16%	0.03%, 0.10%
$E(H)_9$, size of 2×2, $L = 3000$	53.40%, 33.10%	46.57%, 33.43%	1.10%, 4.56%	0.10%, 0.03%
$E(H)_{10}$, size of 2×2, $L = 3000$	26.17%, 31.73%	29.10%, 26.56%	0.03%, 22.46%	31.97%, 0.03%
$E(H)_{11}$, size of 2×2, $L = 3000$	23.67%, 31.40%	26.63%, 30.33%	0.03%, 20.60%	32.77%, 0.03%
$E(H)_{12}$, size of 2×2, $L = 3000$	54.43%, 37.93%	47.00%, 32.80%	1.27%, 3.16%	0.03%, 0.10%
$E(H)_{13}$, size of 2×2, $L = 3000$	26.17%, 31.70%	29.10%, 26.56%	0.03%, 22.46%	31.97%, 0.03%
$E(H)_{14}$, size of 4×4, $L = 1500$	68.37%, 75.63%	67.70%, 73.53%	0.07%, 21.43%	0.03%, 0.03%
$E(H)_{15}$, size of 4×4, $L = 1500$	69.70%, 74.70%	66.43%, 71.83%	0.30%, 0.13%	0.03%, 0.03%
$E(H)_1$, size of 4×4, $L = 1500$	70.67%, 76.46%	67.83%, 73.96%	0.17%, 0.30%	67.63%, 0.03%
$E(H)_{16}$, size of 4×4, $L = 1500$	68.60%, 75.76%	65.17%, 71.73%	0.17%, 0.20%	66.10%, 0.03%
$E(H)_{17}$, size of 4×4, $L = 1500$	68.37%, 75.63%	67.70%, 73.53%	0.07%, 21.43%	0.03%, 0.03%
$E(H)_{18}$, size of 4×4, $L = 1500$	69.70%, 74.70%	66.53%, 71.83%	0.30%, 0.13%	0.03%, 0.03%
$E(H)_{19}$, size of 4×4, $L = 1500$	70.63%, 76.46%	67.77%, 74.13%	0.17%, 0.30%	67.63%, 0.03%
$E(H)_{20}$, size of 4×4, $L = 1500$	68.60%, 75.70%	65.17%, 71.73%	0.17%, 0.20%	66.07%, 0.03%
$E(H)_{21}$, size of 4×4, $L = 1500$	69.70%, 74.70%	66.20%, 71.83%	0.30%, 0.13%	0.03%, 0.03%
$E(H)_{22}$, size of 4×4, $L = 1500$	70.67%, 76.46%	67.77%, 74.13%	0.17%, 0.30%	67.63%, 0.03%
$E(H)_{23}$, size of 4×4, $L = 1500$	68.60%, 75.73%	65.17%, 71.73%	0.17%, 0.20%	66.07%, 0.03%
$E(H)_{24}$, size of 4×4, $L = 1500$	64.57%, 72.53%	63.57%, 71.33%	0.17%, 0.26%	0.03%, 0.03%
$E(H)_{25}$, size of 4×4, $L = 1500$	67.73%, 74.26%	66.40%, 73.50%	0.13%, 0.26%	0.03%, 0.03%
$E(H)_2$, size of 16×16, $L = 375$	**90.60**%, **93.23**%	**90.00**%, **92.46**%	0.20%, 0.30%	0.00%, 0.03%
$E(H)_{26}$, size of 16×16, $L = 375$	90.57%, 92.30%	89.47%, 92.16%	0.23%, 78.53%	0.03%, 0.03%
$E(H)_{27}$, size of 16×16, $L = 375$	89.43%, 92.56%	89.03%, 92.20%	0.13%, 72.03%	0.03%, 0.03%
$E(H)_{28}$, size of 16×16, $L = 375$	74.27%, 82.63%	62.33%, 75.60%	0.87%, 17.50%	0.03%, 0.03%
$E(H)_{29}$, size of 16×16, $L = 375$	74.40%, 82.16%	62.90%, 76.20%	0.17%, 24.76%	0.03%, 0.03%
$E(H)_{30}$, size of 16×16, $L = 375$	77.83%, 84.83%	65.80%, 80.26%	0.20%, 24.20%	0.03%, 0.03%
$E(H)_{31}$, size of 16×16, $L = 375$	77.57%, 83.80%	65.57%, 78.20%	0.23%, 19.30%	0.03%, 0.03%
$E(H)_{32}$, size of 25×25, $L = 240$	87.77%, 91.00%	85.80%, 90.43%	0.37%, 68.46%	87.03%, 90.40%
$E(H)_{33}$, size of 25×25, $L = 240$	87.73%, 91.76%	86.67%, 91.00%	0.40%, 0.86%	0.03%, 90.96%
$E(H)_{34}$, size of 25×25, $L = 240$	88.30%, 91.26%	86.23%, 90.50%	0.33%, 68.63%	87.20%, 91.16%
$E(H)_{35}$, size of 25×25, $L = 240$	87.50%, 91.56%	85.73%, 91.26%	0.07%, 75.16%	0.03%, **91.40**%
$E(H)_{36}$, size of 25×25, $L = 240$	87.20%, 91.00%	85.70%, 90.43%	0.27%, 69.50%	85.87%, 90.7%
$E(H)_{37}$, size of 25×25, $L = 240$	86.67%, 90.80%	86.43%, 90.86%	0.03%, 0.03%	0.03%, 91.03%
$E(H)_{38}$, size of 25×25, $L = 240$	86.50%, 90.60%	85.20%, 89.83%	0.27%, 68.76%	85.00%, 90.33%
$E(H)_{39}$, size of 25×25, $L = 240$	86.83%, 90.73%	86.07%, 89.83%	0.03%, 0.06%	0.03%, 89.86%
$E(H)_{40}$, size of 25×25, $L = 240$	87.37%, 91.56%	85.97%, 90.56%	0.53%, 0.56%	86.63%, 91.00%
$E(H)_{41}$, size of 25×25, $L = 240$	86.93%, 91.03%	85.57%, 89.66%	0.30%, **79.16**%	86.13%, 90.00%
$E(H)_{42}$, size of 48×48, $L = 125$	88.60%, 91.86%	87.73%, 90.46%	0.20%, 78.73%	**87.23**%, 0.03%

$$E_1 = \begin{pmatrix} I_{77} + I_{204} + I_{487} + I_{570} + I_{819} \\ I_{284} + I_{349} + I_{1119} + I_{1149} + I_{1323} \\ 0 \\ 0 \end{pmatrix},$$

$$E_2 = \begin{pmatrix} 0 \\ I_{26} + I_{255} + I_{381} + I_{676} + I_{1049} \\ I_{63} + I_{89} + I_{282} + I_{329} + I_{840} \\ 0 \end{pmatrix},$$

$$E_3 = \begin{pmatrix} 0 \\ 0 \\ I_{188} + I_{301} + I_{797} + I_{1056} + I_{1138} \\ I_{229} + I_{685} + I_{694} + I_{990} + I_{1260} \end{pmatrix},$$

$$E_4 = \begin{pmatrix} I_{41} + I_{108} + I_{111} + I_{940} + I_{945} \\ 0 \\ 0 \\ I_{259} + I_{444} + I_{963} + I_{691} + I_{1002} \end{pmatrix},$$

$$E_5 = \begin{pmatrix} 0 & 0 & I_{109} + I_{328} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & I_1 + I_{355} & 0 & 0 \\ I_{65} + I_{303} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{112} + I_{296} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_1 + I_{166} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I_8 + I_{216} & 0 \\ I_{200} + I_{354} & 0 & I_{65} + I_{267} & 0 \\ 0 & 0 & 0 & I_{49} + I_{74} \\ I_{113} + I_{265} & I_{54} + I_{113} & 0 & 0 \\ I_{10} + I_{374} & I_8 + I_{211} & 0 & I_{238} + I_{244} \\ I_{179} + I_{314} & I_{67} + I_{95} & I_{193} + I_{277} & I_{223} + I_{329} \\ 0 & I_{255} + I_{309} & I_5 + I_{239} & 0 \end{pmatrix},$$

$$E_6 = \begin{pmatrix} I_{90}+I_{159} & I_0+I_{182} & 0 & 0 \\ I_{50}+I_{132} & 0 & I_{78}+I_{186} & I_9+I_{138} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{24}+I_{31} \\ 0 & I_5+I_{24} & 0 & 0 \\ I_{97}+I_{363} & I_{161}+I_{251} & I_{304}+I_{326} & I_{33}+I_{288} \\ I_{17}+I_{347} & 0 & 0 & 0 \\ 0 & 0 & I_{111}+I_{298} & 0 \\ 0 & 0 & I_0+I_{369} & 0 \\ 0 & I_{90}+I_{105} & 0 & 0 \\ I_{27}+I_{368} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{136}+I_{254} \\ 0 & I_{13}+I_{60} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$E_7 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & I_{315}+I_{363} & I_{21}+I_{138} & I_6+I_{153} \\ I_{220}+I_{341} & I_{28}+I_{161} & 0 & 0 \\ 0 & I_{17}+I_{301} & 0 & I_{37}+I_{274} \\ I_{54}+I_{259} & I_{256}+I_{301} & I_{37}+I_{227} & 0 \\ 0 & 0 & I_{254}+I_{307} & 0 \\ 0 & 0 & I_{46}+I_{246} & 0 \\ I_{47}+I_{267} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I_{272}+I_{281} & 0 \\ I_{49}+I_{104} & 0 & 0 & I_{26}+I_{249} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{295}+I_{332} \\ I_{132}+I_{285} & I_{279}+I_{362} & 0 & 0 \\ 0 & I_{18}+I_{112} & 0 & I_{24}+I_{134} \end{pmatrix},$$

$$E_8 = \begin{pmatrix} 0 & 0 & I_{114}+I_{136} & I_{68}+I_{298} \\ 0 & 0 & 0 & I_{149}+I_{173} \\ I_{81}+I_{129} & 0 & 0 & I_{281}+I_{317} \\ 0 & 0 & I_{283}+I_{338} & 0 \\ 0 & 0 & 0 & I_{156}+I_{218} \\ 0 & 0 & 0 & 0 \\ 0 & I_{72}+I_{185} & 0 & 0 \\ 0 & I_{33}+I_{186} & 0 & 0 \\ I_{296}+I_{315} & I_{97}+I_{238} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I_{97}+I_{208} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & I_{253}+I_{348} & I_{198}+I_{316} & I_{95}+I_{126} \\ 0 & 0 & 0 & 0 \\ I_{87}+I_{303} & 0 & I_{59}+I_{71} & 0 \\ 0 & I_{22}+I_{317} & 0 & I_{18}+I_{112} \end{pmatrix}.$$

FUNDING

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. L. Dall'Amico, R. Couillet, and N. Tremblay, J. Mach. Learn. Res. **22** (217), 1 (2021). https://www.jmlr.org/papers/v22/20-261.html.
2. J. Lang, S. Sachdev, and S. Diehl, arXiv Preprint (2024). https://doi.org/10.48550/arXiv.2406.05842
3. M. Talagrand, *Mean Field Models for Spin Glasses, Vol. II: Advanced Replica-Symmetry and Low Temperature*, 2nd ed., Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge/A Series of Modern Surveys in Mathematics, Vol. 55 (Springer, 2011). https://doi.org/10.1007/978-3-642-22253-5
4. M. Fiedler, Czech. Math. J. **23**, 298 (1973). https://doi.org/10.21136/cmj.1973.101168
5. K. Hayashi, S. G. Aksoy, Ch. H. Park, and H. Park, in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Association for Computing Machinery, New York, 2020), p. 495. https://doi.org/10.1145/3340531.3412034
6. H. Nishimori, Prog. Theor. Phys. **66**, 1169 (1981). https://doi.org/10.1143/PTP.66.1169
7. L. Dall'Amico, R. Couillet, and N. Tremblay, J. Stat. Mech.: Theory Exp. **2021**, 093405 (2021). https://doi.org/10.1088/1742-5468/ac21d3
8. H. Kwon, J. Jang, J. Kim, K. Kim, and K. Sohn, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024), p. 17364. https://openaccess.thecvf.com/content/CVPR2024/html/Kwon_Improving_Visual_Recognition_with_Hyperbolical_Visual_Hierarchy_Mapping_CVPR_2024_paper.html.
9. S. Ramasinghe, V. Shevchenko, G. Avraham, and A. Thalaiyasingam, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024), p. 27263. https://openaccess.thecvf.com/content/CVPR2024/html/Ramasinghe_Accept_the_

Modality_Gap_An_Exploration_in_the_Hyperbolic_Space_CVPR_2024_paper.html.

10. K. Simonyan and A. Zisserman, in *3rd International Conference on Learning Representations (ICLR 2015)* (2015), p. 1.
11. K. He, X. Zhang, Sh. Ren, and J. Sun, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, 2016* (IEEE, 2016), p. 770. https://doi.org/10.1109/CVPR.2016.90
12. M. Tan and Q. Le, Proc. Mach. Learn. Res. **97**, 6105 (2019).
13. Z. Liu, Yu. Lin, Yu. Cao, H. Hu, Yi. Wei, Zh. Zhang, S. Lin, and B. Guo, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, 2021* (IEEE, 2021), p. 10012. https://doi.org/10.1109/iccv48922.2021.00986
14. W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, Comput. Visual Media **8**, 415 (2022). https://doi.org/10.1007/s41095-022-0274-8
15. J. S. Wang, W. Selke, V. S. Dotsenko, and V. B. Andreichenko, Europhys. Lett. **11**, 301 (1990). https://doi.org/10.1209/0295-5075/11/4/002
16. I. A. Gruzberg, N. Read, and A. W. W. Ludwig, Phys. Rev. B **63**, 104422 (2001). https://doi.org/10.1103/physrevb.63.104422
17. J. Xie, Y. Lu, S.-C. Zhu, and Y. Wu, Proc. Mach. Learn. Res. **48**, 2635 (2016). https://proceedings.mlr.press/v48/xiec16.html.
18. M. Hill, J. C. Mitchell, and S.-C. Zhu, in *International Conference on Learning Representations* (2021). https://openreview.net/forum?id=gwFTuzxJW0.
19. P. Erdős and A. Rényi, Publications of the Mathematical Institute of the Hungarian Academy of Sciences **5**, 17 (1960).
20. M. P. C. Fossorier, IEEE Trans. Inf. Theory **50**, 1788 (2004). https://doi.org/10.1109/TIT.2004.831841
21. T. J. Richardson and R. L. Urbanke, in *Bob McEliece's Workshop* (2002).
22. T. Richardson, in *Proceedings of the 41st Annual Allerton Conference on Communication, Control and Computing* (2003), p. 1426.
23. V. S. Usatyuk and S. I. Egorov, in *2024 26th International Conference on Digital Signal Processing and its Applications (DSPA), Moscow, 2024* (IEEE, 2024), p. 1. https:doi.org/10.1109/DSPA60853.2024. 10510058
24. T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, IEEE Trans. Commun. **52**, 1242 (2004). https://doi.org/10.1109/TCOMM.2004.833048
25. D. Vukobratovic, A. Djurendic, and V. Senk, in *2007 IEEE International Conference on Communications, Glasgow, 2007* (IEEE, 2007), p. 665. https://doi.org/10.1109/ICC.2007.114
26. K. Deka, A. Rajesh, and P. K. Bora, in *2014 8th International Symposium on Turbo Codes and Iterative Information Processing (ISTC), Bremen, Germany, 2014* (IEEE, 2014), p. 67. https://doi.org/10.1109/ISTC.2014.6955087
27. H. Huang and Yo. Kabashima, Phys. Rev. E **90**, 052813 (2014). https://doi.org/10.1103/PhysRevE.90.052813
28. D. J. Costello and G. D. Forney, Proc. IEEE **95**, 1150 (2007). https://doi.org/10.1109/jproc.2007.895188
29. Ch. Poulliat, M. Fossorier, and D. Declercq, IEEE Trans. Commun. **56**, 1626 (2008). https://doi.org/10.1109/TCOMM.2008.060527
30. X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, IEEE Trans. Inf. Theory **51**, 386 (2005). https://doi.org/10.1109/TIT.2004.839541
31. V. S. Usatyuk, Progressive Edge Growth for LDPC Code Construction (GitHub). https://github.com/Lcrypto/classic-PEG-.
32. V. Usatyuk and I. Vorobyev, in *2018 41st International Conference on Telecommunications and Signal Processing (TSP), Athens, 2018* (IEEE, 2018), p. 1. https:// doi.org/10.1109/TSP.2018.8441303
33. V. S. Usatyuk and S. I. Egorov, in *2024 26th International Conference on Digital Signal Processing and its Applications (DSPA), Moscow, 2024* (IEEE, 2024), p. 1. https:// doi.org/10.1109/DSPA60853.2024.10510073
34. V. S. Usatyuk and D. A. Sapoznikov, RBIM Graph Clustering (GitHub). https://github.com/Lcrypto/Classical-and-Quantum-Topology-ML-toric-spherical/tree/main/RBIM_Nishimori_Clustering.
35. E. Fix and J. L. Hodges, Jr., Int. Stat. Rev. **57**, 238 (1989).
36. T. Cover and P. Hart, IEEE Trans. Inf. Theory **13**, 21 (1967). https://doi.org/10.1109/TIT.1967.1053964
37. Yu. Watanabe and K. Fukumizu, in *Advances in Neural Information Processing Systems*, Ed. by Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (Curran Associates, 2009), Vol. 22, p. 2017. https://proceedings.neurips.cc/paper_files/paper/

2009/file/b6f0479ae87d244975439c6124592772-Paper.pdf.

38. Y. Watanabe and K. Fukumizu, arXiv Preprint (2011). https://doi.org/10.48550/arXiv.1103.0605
39. I. Sato, H. Mitsuhashi, and H. Morita, Linear Multilinear Algebra **62**, 114 (2014). https://doi.org/10.1080/03081087.2013.764496
40. D. J. MacKay and M. C. Davey, in *Codes, Systems, and Graphical Models*, The IMA Volumes in Mathematics and its Applications, Vol. 123 (Springer, New York, 2001), p. 113. https://doi.org/10.1007/978-1-4613-0165-3_6
41. R. Smarandache and P. O. Vontobel, IEEE Trans. Inf. Theory **58**, 585 (2012). https://doi.org/10.1109/tit.2011.2173244
42. I. C. Duta, M. I. Georgescu, and R. T. Ionescu, in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, 2021* (IEEE, 2021), p. 403. https://doi.org/10.1109/ICCVW54120.2021.00049
43. J. Liu, X. Zhang, and Zh. Luo, Entropy **26**, 469 (2024). https://doi.org/10.3390/e26060469

AI tools may have been used in the translation or editing of this article.